DSO 562 - Group 8

Product Applications:
Supervised Identity Fraud Detection

Team Members:
Divya Sripathy
Fabbiha Islam
Hongxuan Wang
Jayant Maheshwari
Jude Alfuraih
Shashank Tiwari
Snehil Saraswat

March 15, 2020

Table of Contents

Executive Summary

Group 8 has been tasked with building a supervised fraud model to detect identity fraud in real-time for product applications. The following report provides a detailed description of how the group constructed and implemented an algorithm to identify identity fraud in the applications data provided. The data consists of 1,000,000 records and 10 fields, including existing fraud labels, which allowed the group to build a robust model capable of efficiently identifying fraudulent applications.

To implement the model, the group proceeded with the following steps:

- **Description of Data:** Overview of the most significant fields and their distributions.
- **Data Cleaning:** Treatment of frivolous values and filling leading zeros.
- **Candidate Variables:** Creation of 253 candidate variables, from the original 8 fields.
- **Feature Selection:** Z-Scaling, Univariate KS and FDR filters, and RFECV wrapper to identify the 27 best variables.
- **Model Algorithms:** Implementation of the base linear model (Logistic Regression), and 3 non-linear models (Boosted Tree, Random Forest, and Neural Network) to calculate average FDR at 3%.
- **Results:** Final model and parameter selection to create three statistics tables for training, testing, and OOT.

Our high-level results are as follows:

	Average FDR @ 3%		
	Training	**Testing**	**Out of Time**
Logistic Regression	0.496	0.479	0.475
Random Forest	0.518	0.508	0.505
Boosted Tree	0.534	0.524	0.520
Neural Network	0.535	0.526	0.525

Based on the above results, Neural Network was chosen as our final model as it has the best performance for FDR at 3% for the testing and out of time datasets. A list of 3 statistics table for training, testing and OOT has also been provided in the results section of the report.

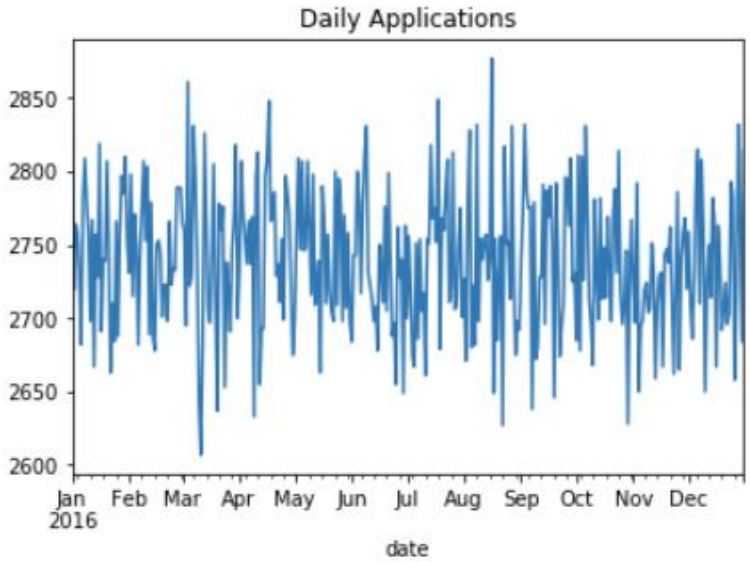
Data Description

The Product Applications dataset consists of synthesized product application data from the year 2016. The dataset contains 10 columns and 1 million records, including date (of application), SSN (Social Security Number), firstname (applicant's first name), lastname (applicant's last name), address, zip5 (5-digit zip code), dob (date of birth), homephone (9-digit phone number including area code) and fraud labels (1 for fraudulent and 0 for non-fraudulent). The applications dataset did not contain missing values for any of the fields, however frivolous values were present and were treated before further analysis was conducted.

Field Name	# of Records	% Populated	# Unique Values	Most Common Field Value
record	1000000	100%	1000000	NA
date	1000000	100%	365	20160816
ssn	1000000	100%	835819	999999999
firstname	1000000	100%	78136	EAMSTRMT
lastname	1000000	100%	177001	ERJSAXA
address	1000000	100%	828774	123 MAIN ST
zip5	1000000	100%	26370	68138
dob	1000000	100%	42673	19070626
homephone	1000000	100%	28244	999999999
fraud_label	1000000	100%	2	0

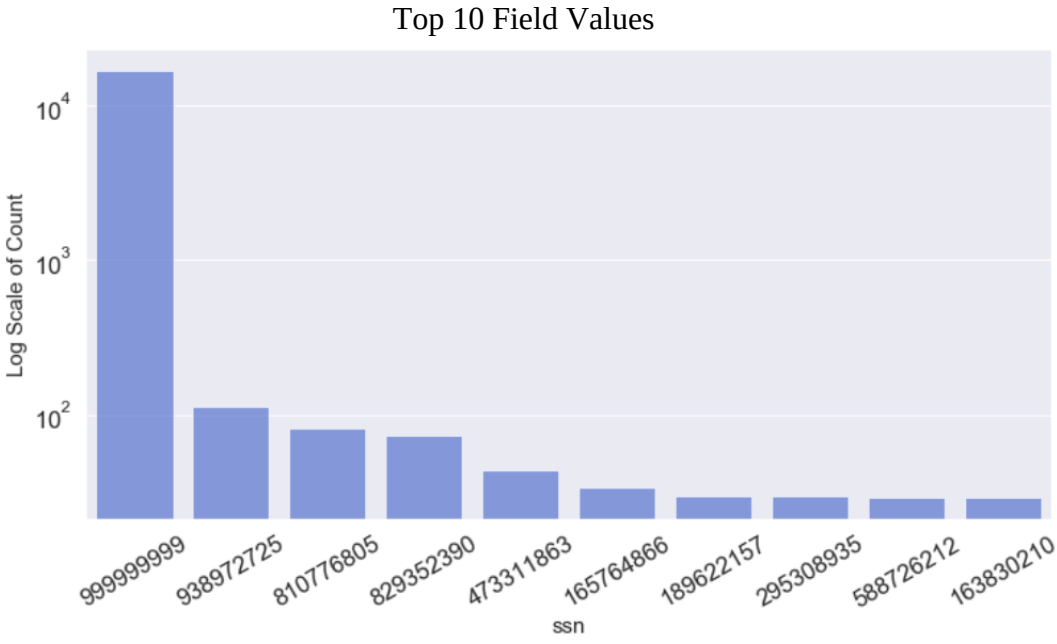
A brief description of the significant fields can be found below. A full detailed description of all fields in the dataset is provided in Appendix 1.1.

1. Field Name: date
Date is a categorical variable that denotes the date on which the application was submitted. It has 365 unique values (February 29th data is not present) ranging from January 1, 2016 to December 31, 2016. This field does not contain missing values. To plot a more accurate distribution, the daily count of applications for '2016-02-29' has been made equivalent to the daily count of applications for '2016-02-28' for consistency.



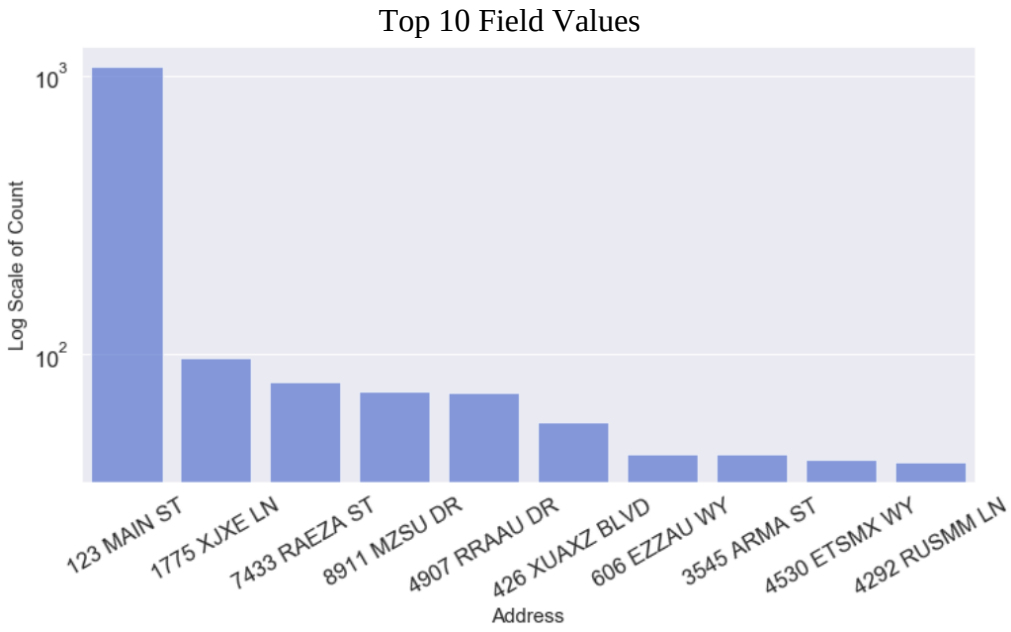
2. Field Name: ssn

SSN is a categorical field that denotes the social security number of the applicant. It has 835,819 unique values and no missing records. The distribution of the top 10 values shows that '999999999' has an abnormally high occurrence compared to other values. This is considered a frivolous value as it could be a potential default value that was used when the information was not provided for the given application.



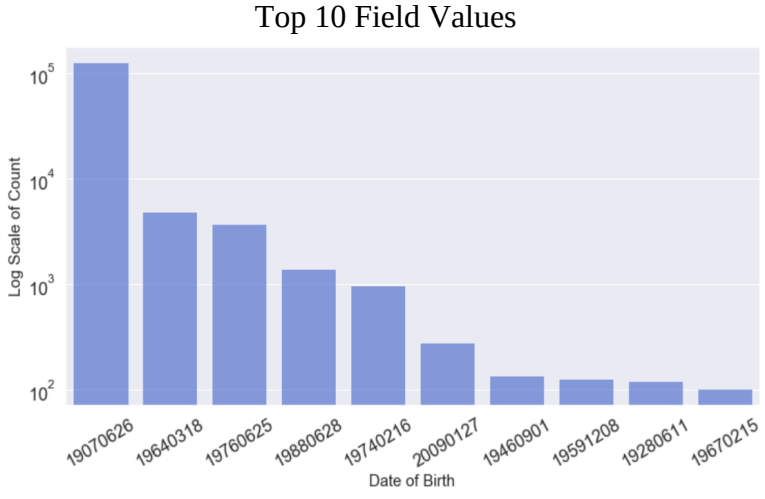
3. Field Name: address

Address is a categorical field that denotes the address provided by the applicant. It contains 828,774 unique values and has no missing records. The distribution of the top 10 values shows that '123 MAIN ST' has an abnormally high occurrence compared to other values. Given its generic nature and the likelihood that it was a potential default value for missing data, it should be treated as a frivolous value.



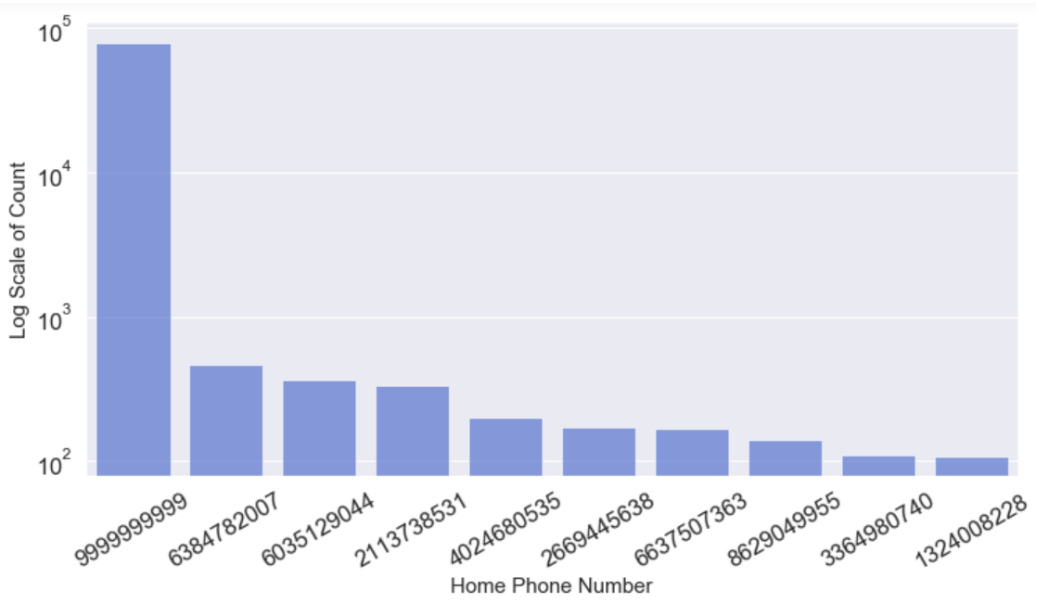
4. Field Name: dob

DOB is a date field that denotes the date of birth of the applicant. This field contains 42,673 unique values with no missing records. The distribution of the top 10 values shows that '19070626' has an abnormally high occurrence compared to other values. Given its generic nature and the likelihood that it was a potential default value for missing data, it should also be treated as a frivolous value.



5. Field Name: homephone

Homephone is a categorical field that denotes the 10-digit home phone number of the applicant, including area code. It takes on 28,244 unique values and does not have any missing records. The distribution of the top 10 values shows that '999999999' has an abnormally high occurrence compared to other values. Given its generic nature and the likelihood that it was a potential default value for missing data, it should be treated as a frivolous value.



6. Field Name: fraud_label

Fraud_label is a categorical field that classifies the record as fraudulent or non-fraudulent. It contains two unique values: "0" for non-fraudulent and "1" for fraudulent. This field has no missing records. There are 985,607 records labeled as non-fraudulent, and 14,393 records labeled as fraudulent.

Fraud Label	Count
0	985607
1	14393

Data Cleaning

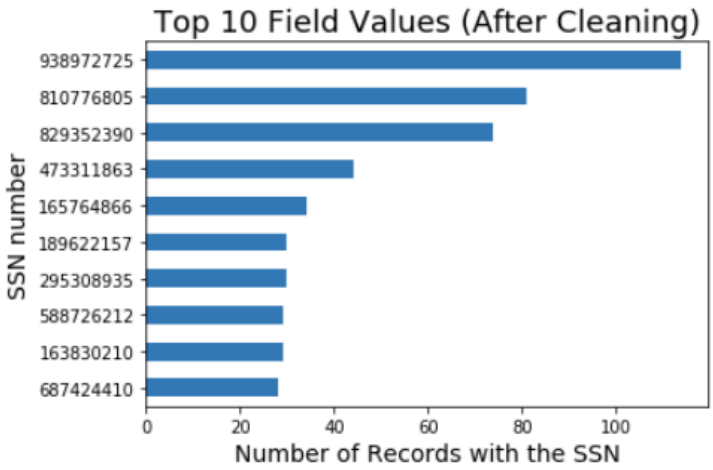
After exploring and understanding the data provided and its important elements, the next major step in any machine learning algorithm is data cleaning. The above data was initially checked for the presence of null values to ensure that all fields were fully populated. Data imputation methods were not needed as no missing values were found in the dataset.

The most significant portion of data cleaning concerning this model involved treating frivolous values in the data, as identified above in the data description. Frivolous values are those that hold little to no significance for prediction and could be misleading for the supervised learning algorithm when identifying fraud. They are often the default place fillers for missing data. To identify these frivolous values, bar graphs were plotted to visualize the frequency of each unique value for the fields in the dataset. It was found that the fields “ssn”, “address”, “homephone” and “dob” contained frivolous values.

These frivolous values were treated by replacing each instance with the record number for its corresponding row. The reason behind this approach is that a unique value for this field will not be detected by a fraud algorithm (which looks for repeat occurrences of each field value).

Outlined below are the frivolous values that were treated, and the resulting plots of value counts after treatment:

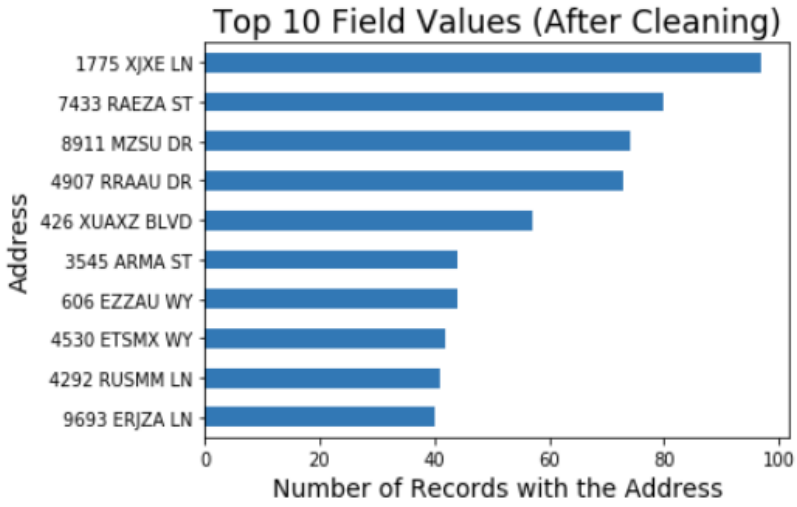
1. Field: “ssn”
Frivolous Value: “999999999”
Treatment: For records containing “9999999999” in the social security number field, the value was replaced with the corresponding value in the “record” field.



2. Field: “address”

Frivolous Value: “123 MAIN ST”

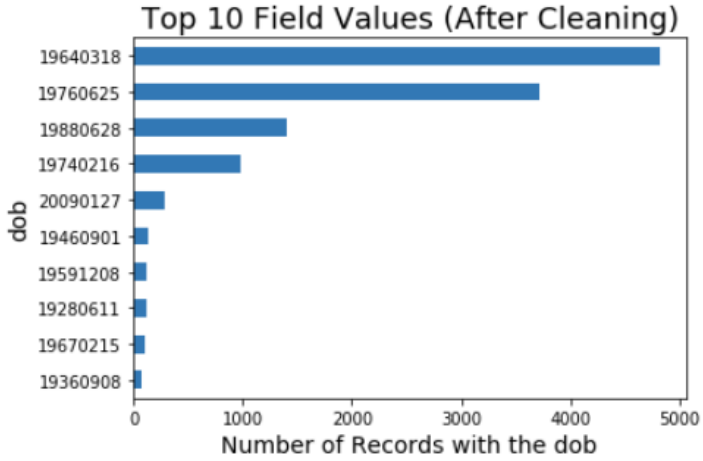
Treatment: For records containing “123 MAIN ST” in the address field, the value was replaced with the value in the “record” field.



3. Field: “dob”

Frivolous Value: “19070626”

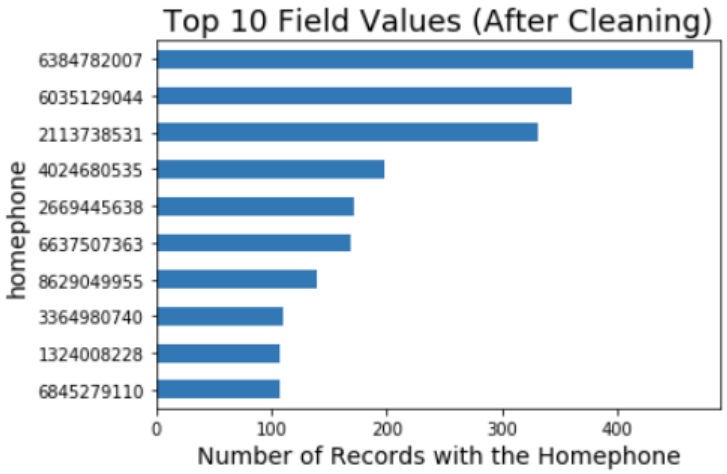
Treatment: For records containing “19070626” in the date of birth field, the value was replaced with the value in the “record” field.



4. Field: “homephone”

Frivolous Value: “9999999999”

Treatment: For records containing “9999999999” in the home phone field, the value was replaced with the value in the “record” field.



After treating the frivolous values, the next step was to make the values in these fields consistent. The raw data in the fields “homephone”, “zip5”, “ssn”, and “dob” contained variations in length; for example, phone numbers typically consist of 10 numerical digits, however, the data contained some records with only 9 digits entered. For such cases, it can be assumed that there are leading 0’s to fill in the missing number of digits. Thus, all the aforementioned fields were standardized in length using leading 0’s and had the following final lengths: “homephone” - ten digits, “zip5” - five digits, “ssn” - nine digits, and “dob” with eight digits.

This concluded the data cleaning process for this project. The cleaned data was then ready to be used for the creation of the candidate variables.

Candidate Variables

In order to create optimal variables for the models, a systematic approach was used to create 253 new variables from the original 8 fields to serve as potential inputs to the model.

A total of 18 baseline combination entities were created from our original 8 fields to not only uniquely identify individuals, but also track slight variations in combinations that a fraudster might attempt over the course of multiple fraudulent application attempts. A detailed description of the logic used to create these variables is listed below.

Original 8 fields, as defined in the data description, are as follows:

- date
- ssn
- address
- firstname
- lastname
- zip5
- dob
- Homephone

The approach to determining a list of candidate variables involved the following steps:

1. **Of the initial fields, retain the ones that are most salient to an individual's identity.** The 2 fields that fulfill this criterion are "ssn" and "homephone", since ssn uniquely identifies an individual, and home phone number, while not as specific, can also uniquely identify an applicant. Fields such as first name and last name are not as powerful to identify specific individuals since the same names can be shared among multiple individuals.

2. **Of the available initial fields, create as many possible combinations that can better uniquely identify a person.**
   These combinations help in the following way:
   a. Identifying individuals with specific combinations of identification elements to track their records.
   b. Provide valuable inputs to a machine learning model that can accurately predict a fraud label based on small variations in these more unique traces of identity, since fraudsters who manipulate identities are more likely to change just a few elements of their baseline identity information. A higher number of combinations of identity elements provides better training to the machine learning model to catch these minor variations.

The baseline combination entities created with this approach in mind are as follows:

1. Name (firstname + lastname)
2. Name_DOB (Name + dob)
3. FullAddress (address + zip5)
4. SSN_FullAddress (ssn + FullAddress)
5. SSN_Name_DOB (ssn + Name_DOB)
6. SSN_Phone (ssn + homephone)
7. Name_DOB_FullAddress (Name_DOB + FullAddress)
8. FullAddress_Phone (FullAddress + homephone)
9. Name_DOB_Phone (Name_DOB + homephone)
10. FirstName_SSN (firstname + ssn)
11. LastName_SSN (lastname + ssn)
12. Name_phone (Name + homephone)
13. Name_FullAddress (Name + FullAddress)
14. Name_ZIP (Name + zip5)
15. Name_Phone_FullAddress (Name_phone + FullAddress)
16. Name_SSN (Name + SSN)

The above list, in addition to the 2 variables ssn and homephone, forms the baseline 18 unique entities upon which the candidate variables were built.

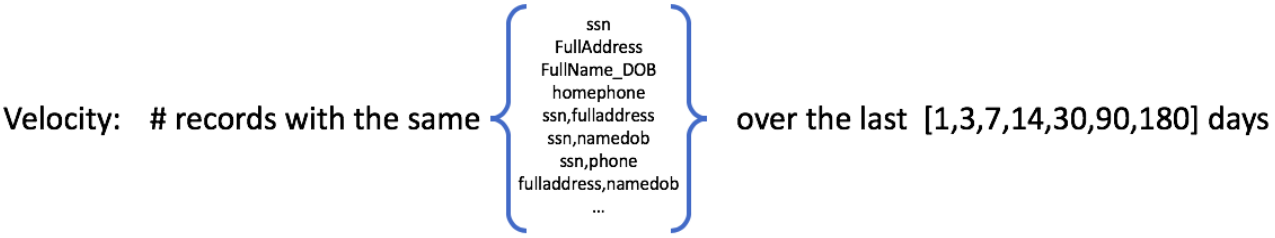
The additional 253 variables created fall primarily into the following 3 categories:

Category 1: Velocity (126 variables)

Velocity refers to the number of records with the combination in question that have been seen over the past *n* days. A list of 7 values of *n* was chosen and calculated for each of the 18 baseline entities:

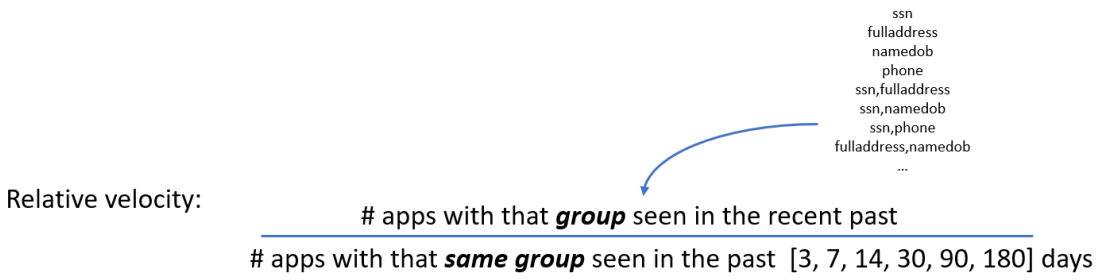
$$n = [1,3,7,14,30,90,180] \quad \text{days}$$

The velocity calculation is diagrammed below:



Category 2: Relative Velocity (108 variables)

Relative velocity refers to the velocity of an entity in the recent past compared to that entity's velocity in the past n days. It can be calculated by dividing the number of each variable found over the last 1 day by the number of applications with that entity seen in different time windows (3 days, 7 days, etc.) and is diagrammed below:

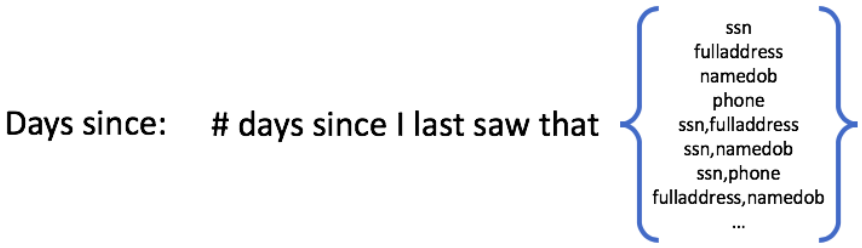


Category 3: Days Since Last Seen (18 variables)

Days Since Last Seen refers to the number of days since a record with the same combination of identity fields was last detected. For example, the name "Thomas_Williams" will have a value of 2 if it was last seen two days ago. For each of the above 18 entities, the days since last seen variables were created, resulting in 18 additional variables, one for each combination.

Besides, all the records that were seen for the first time were filled by the difference of the record's date to a base date (January 1, 2016 in this scenario) as it is the best estimate available for the last seen. To avoid incorrectly flagging records in the first 20 days of the year, that are showing up for the first time as potential frauds, records were set to a minimum value of 20, based on the above logic.

This approach is diagrammed below:



The lower value in category 3 (days since last seen) indicates a higher chance of fraud, whereas, for category 1 (velocity over n days) and category 2 (relative velocity) it is the opposite case (i.e. a lower value indicates a lower chance of fraud). Thus, each of the 18 "days since" variables has been subtracted from its maximum to keep its interpretation consistent with the other 2 categories.

Risk Variable:

A risk variable was also created for the day of week using the date field. Risk variables are the product of target encoding a categorical variable (in this case, day of week). For each day of week, the average fraud value was computed for the risk variable, which is summarized in the below risk table:

Weekday	Weekday Risk
Monday	0.01348
Tuesday	0.01407
Wednesday	0.015169
Thursday	0.014981
Friday	0.014499
Saturday	0.014968
Sunday	0.013674

When summed up, the total number of candidate variables created are 253.

The reason for choosing the above 3 categories to create candidate variables is that fraudsters often submit multiple applications within a short time period, with minor variations in identities. Therefore, by having a large variety of combinations and entities, as well as tracking the days since last seen, velocity and relative velocity for each entity, a machine learning model will likely be able to detect nuances in variations submitted by a potential fraudster multiple times in a time period, and flag the appropriate records as fraudulent.

A list of all the candidate variables that were created can be found in Appendix 1.2 section.

At this point in the analysis, all the 253 candidate variables are on different scales. Hence, z-scaling the variables would help in standardizing the data *before* dividing it into training, testing, and out of time (OOT).

Z-Scaling the Variables

Z-scaling was achieved by subtracting row values from the column mean for each of the variables and dividing it by the standard deviation of the column. This step results in all fields existing on a similar scale with a mean of 0 and a standard deviation of 1.

The Z-Scaling can be done as follows:

$$z = \frac{x - \mu}{\sigma}$$

where:
x is the row value for a given column,
μ is the mean of the column,
σ is the standard deviation of the column.

Once the data was standardized, the OOT validation sample (November to December data) was separated before applying feature selection methods, to act as a source of unseen test data outside the time range of training data for the model.

Feature Selection

After all the candidate variables are built, it is important to select the most important features as it enables the machine learning models to train faster, reduces its complexity, and makes it easier to interpret. Also, the phenomenon of 'curse of dimensionality' arises when analyzing data in high-dimensional spaces, specifically the issue of sparsity and closeness of data and thus, it is important to keep minimal dimensions to find the surface that best fits through the given data points.

The feature selection process involves using a filter and wrapping method to narrow down the variables.

Filter

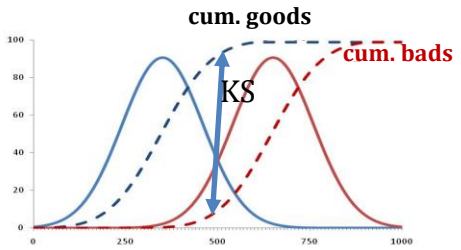
A filter is a variable score that helps in removing the majority of variables that are not effective in differentiating between "goods" (non-fraudulent applications) and "bads" (fraudulent applications). The two popular measures of goodness for fraud are discussed below:

1) Kolmogorov-Smirnov (KS) Test

KS distance is generally used as a guide for finding the maximum separation between two cumulative distribution functions for the response classes (here fraudulent vs. non-fraudulent records).

Formula for KS Distance

$$KS = \max_{x} \int_{x_{min}}^{x} [P_{\text{good}} - P_{\text{bad}}]\, dx$$



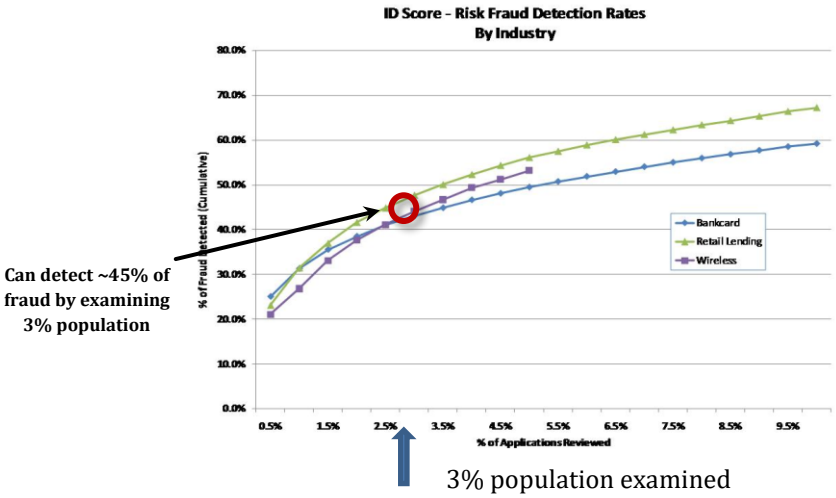
For all the predictor variables in the dataset, Univariate KS distance was calculated on the train/test dataset (initial weeks of January were removed as they were not completely developed), and the variables were then sorted and ranked from highest to lowest value based on the KS distance, i.e. the highest KS distance score was assigned the highest rank. We also added two new variables 'fraud' & 'random' to check our KS scoring. After the score was sorted, the 'fraud' variable was at the top of our list and 'random' was lower down in the list, indicating the correctness of our KS calculation method.

2) Fraud Detection Rate (FDR)

The FDR is a more robust univariate model performance measure that is independent of any modeling methods. It demonstrates the percentage of all frauds that are caught, given a specified examination cut-off point for a variable, usually set by businesses as per their requirement. For

example, FDR 45% at 3% means the model catches 45% of all the frauds in 3% of the population.

Example of FDR



For all the predictor variables in the dataset, FDR at 3% was calculated on the train/test dataset, and the variables were then sorted and ranked from highest to lowest value based on the FDR, i.e. the highest value was assigned the highest rank. We also added two new variables 'fraud' & 'random' to check our FDR's accuracy. After the score was sorted, the 'fraud' variable was at the top of our list and 'random' was lower down in the list, indicating the correctness of our FDR calculation method.

Using KS and FDR rankings, an average score was calculated for all the variables and only the top 50% of the variables were selected for the next stage to feed to the wrapper model.

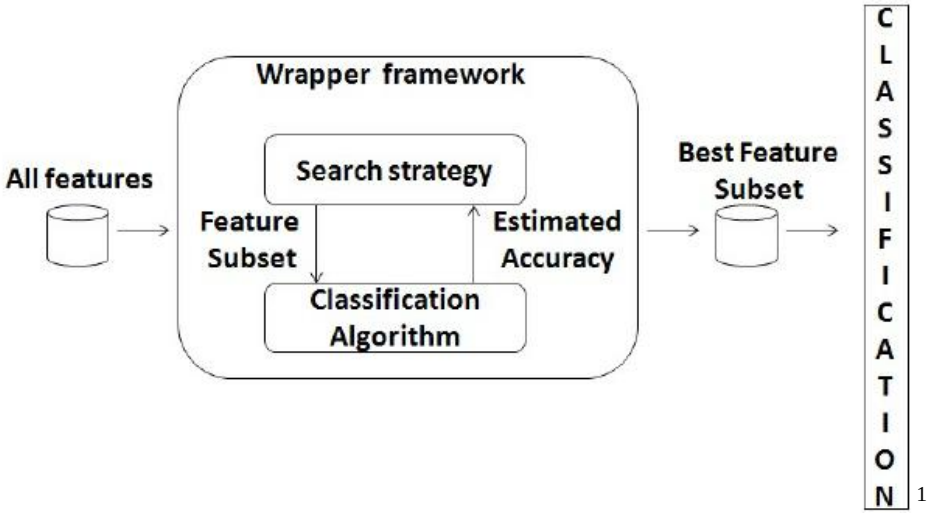
Wrapper

A wrapper method wraps a model around the process. It is of mainly 3 types:

- *Forward Selection* - Variables are added till the point no substantial improvement is seen.
- *Backward Selection* - Variables are removed until the model degradation is below an acceptable amount.
- *General Stepwise Selection* – Variables can be added or removed at any stage.

For this project, a backward elimination-based wrapper method known as Recursive Feature Elimination with Cross-Validation (RFECV) is used to further reduce our number of variables. This method starts with creating a model (logistic regression in this case) for all the input variables and then removes the least important predictors in each step based upon the importance score of each variable calculated by the model after cross-validation. RFECV can be time-

consuming and thus an expensive task. Through multiple iterations of the wrapper, the 27 ‘best variables’ (available in Appendix 1.3) to train the classification models were chosen. The following figure illustrates this process:



[1]

[1] https://www.researchgate.net/figure/The-process-of-wrapper-feature-selection_fig4_271513838

Model Algorithms

It is imperative to divide the data into training and testing for model training and validation purposes before building models. This is done in the following way:

First, an out-of-time partition was made in the dataset for all records after November 1st, 2016. This out-of-time dataset is used to predict how the model will perform on data it has never seen before in a new time range, i.e. to evaluate how the model will perform on data as it arrives over time to help identify which transactions are fraudulent.

With the remaining data from before November 1, 2016, a training/testing split of 0.70 was carried out, meaning 70% percent of the data was randomly chosen as training data and the remaining 30% as the testing data.

Using the 27 final variables that were identified, five supervised machine learning models to predict the probability of a record being fraud were created. A simple Logistic Regression model was used as the baseline, after which three non-linear models - Random Forest, Boosted Trees, and Neural Network - were built to evaluate the improvement of model performance.

Baseline Linear Model

The first step to approach a supervised learning problem is to try a linear model to establish a baseline comparison benchmark, before trying other non-linear models such as random forest, XGBoost, etc.

Baseline models are simple linear classifiers that are quicker to run and can reach a high level of accuracy. They also allow for a better understanding of the outputs and behavior of a non-linear model, are more interpretable, and thus provide a good comparison point when running non-linear models. In the case that subsequent non-linear models can't beat the baseline metrics, the simple linear model should be the optimal option.

In this supervised fraud detection problem, the baseline linear model used was logistic regression: a linear classifier often used for supervised binary classification. Logistic regression models the probability that a record is likely to belong to a class of interest, and uses the sigmoid function as seen below to assign probability values between 0 and 1:

$$\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right) = \beta_0 + \beta_1 x$$

,

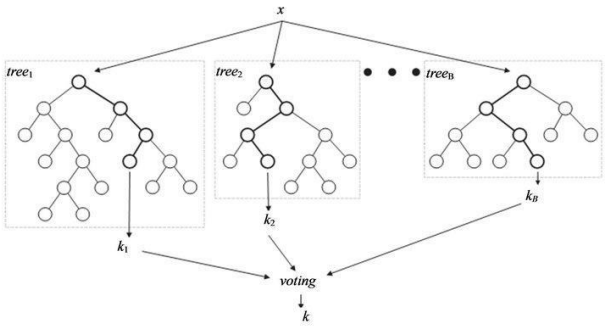
where the optimal parameters β0 and β1 are those that maximize the likelihood function.

Non-linear Models

After creating the linear model, we have a benchmark that we try to improve upon using non-linear models. These are more complex in general but are capable of giving better results.

- *Random Forest* - Random Forest is a powerful ensemble method, which consists of a set of decision tree models built with a subset of all variables. With each decision tree, the node represents a predictor, and the leaf represents the number of samples in each class. When building trees in Random Forest, each time we consider a subset of variables and use that subset to grow a tree and obtain a prediction result. By averaging or voting the results, the algorithm can achieve better prediction accuracy than a single decision tree which is more susceptible to overfitting. Typically, the number of variables considered at each split is approximately equal to the square root of the total number of variables. Important parameters for Random Forest are the number of trees, maximum depth of each tree, and minimum number of samples allowed in each leaf.

Random Forest Method



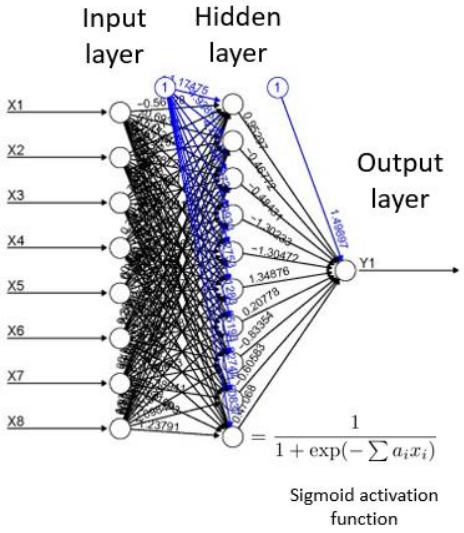
- *Boosted Tree* - Boosted Tree is another powerful ensemble method which combines several weak learners to build a single strong learner. The Boosted Tree algorithm is based on the idea that by building a series of trees in sequence, prediction accuracy can be improved by learning from the mistakes of the previous tree. A Boosted Tree is built by training many "weak" learners in sequence. A weak learner is a constrained model (i.e. you could limit the maximum depth of each decision tree). Each one in the sequence focuses on learning from the mistakes of the one before it. Boosting then combines all the weak learners into a single strong learner.

Boosted Trees Method

- *Neural Network* - Neural Network is a computing algorithm that is emulated from biological neural nets with a neuron as its fundamental unit. They are used for solving machine learning problems dealing with calculating outputs based on patterns in the bulk of the data. It does so by learning on the data inputted to a net. They interpret

sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Neural Network Method



- *Support Vector Machine* - A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space, this hyperplane is a line dividing a plane into two parts, where each class lies on either side. Support Vector Machine is an extension of the Support Vector classifier that results from enlarging the feature space in a specific way using kernels.

SVM Method

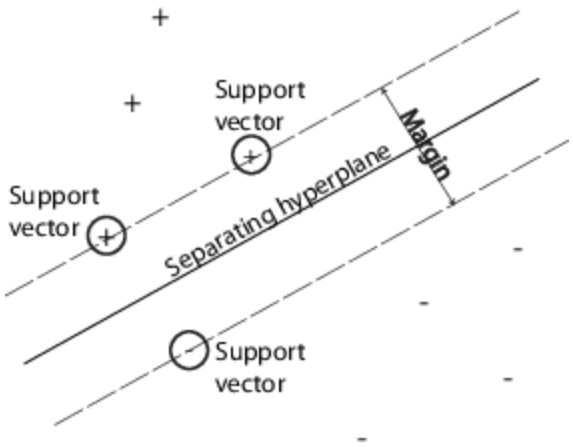


Image Credit: MathWorks

Due to the long processing time for SVM, only the first 4 models were trained and tested for the final 27 variables.

Model Results

For each model, the performance is evaluated by comparing the Fraud Detection Rate (FDR) at 3%, i.e., the percentage actual fraudulent transactions identified in the top 3% of the population based on the decreasing score of each transaction being fraudulent according to the model.
The Fraud Detection Rate at 3% was calculated for all the models for training, testing and out-of-time data. Below is a summary of the model performance:

	Average FDR @ 3%		
	Training	**Testing**	**Out of Time**
Logistic Regression	0.496	0.479	0.475
Random Forest	0.518	0.508	0.505
Boosted Tree	0.534	0.524	0.520
Neural Network	0.535	0.526	0.525

Based on the above results, Neural Network was chosen as our final model as it has the best performance for FDR at 3% for testing and Out of Time dataset.

Results

Based on the above comparison, Neural Network is selected as the final model. Below charts show a detailed breakdown of the model performance in training, testing and Out-of-Time data. The steps to obtain the breakdown is:

a. All the records were sorted in a decreasing order of their likelihoods of being fraudulent.
b. The entire ordered population is then divided into 100 bins of equal size. We look at only the top 20 bins of interest. Some bins may contain 1 more record than the other, since the number of records in each bin must be an integer.
c. Calculate the number of "Good's" (non-fraudulent) and the number of" Bad's" (fraudulent) in each of the 20 bins.
d. Calculate the percentage of "Good's" (non-fraudulent) and the percentage of "Bad's" (fraud) in each of the 20 bins.
e. After calculating this information for each bin, calculate the cumulative number and percentage of "Good's" and "Bad's" out of all the "Good's" and "Bad'" in the dataset from bins 1 to 20.
f. KS is defined as the maximum separation of two distributions. Based on this definition, we calculate the KS by subtracting the percentage of "Good's" from the "Bad's".
g. We calculate the False Positive Rate (FPR) by dividing the Cumulative Good by Cumulative Bad.

Training Statistics for Neural Network

Training	# Records	# Goods	# Bads	Fraud Rate
	583454	575111	8343	0.014299328

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	%Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	5834	1590	4244	27.25%	72.75%	5834	1590	4244	0.28%	50.87%	50.59	0.37
2	5834	5694	140	97.60%	2.40%	11668	7284	4384	1.27%	52.55%	51.28	1.66
3	5834	5785	49	99.16%	0.84%	17502	13069	4433	2.27%	53.13%	50.86	2.95
4	5834	5793	41	99.30%	0.70%	23336	18862	4474	3.28%	53.63%	50.35	4.22
5	5834	5793	41	99.30%	0.70%	29170	24655	4515	4.29%	54.12%	49.83	5.46
6	5834	5797	37	99.37%	0.63%	35004	30452	4552	5.29%	54.56%	49.27	6.69
7	5834	5796	38	99.35%	0.65%	40838	36248	4590	6.30%	55.02%	48.71	7.90
8	5834	5793	41	99.30%	0.70%	46672	42041	4631	7.31%	55.51%	48.20	9.08
9	5834	5788	46	99.21%	0.79%	52506	47829	4677	8.32%	56.06%	47.74	10.23
10	5834	5784	50	99.14%	0.86%	58340	53613	4727	9.32%	56.66%	47.34	11.34
11	5834	5799	35	99.40%	0.60%	64174	59412	4762	10.33%	57.08%	46.75	12.48
12	5834	5793	41	99.30%	0.70%	70008	65205	4803	11.34%	57.57%	46.23	13.58
13	5834	5789	45	99.23%	0.77%	75842	70994	4848	12.34%	58.11%	45.76	14.64
14	5834	5797	37	99.37%	0.63%	81676	76791	4885	13.35%	58.55%	45.20	15.72
15	5834	5794	40	99.31%	0.69%	87510	82585	4925	14.36%	59.03%	44.67	16.77
16	5834	5787	47	99.19%	0.81%	93344	88372	4972	15.37%	59.59%	44.23	17.77
17	5834	5792	42	99.28%	0.72%	99178	94164	5014	16.37%	60.10%	43.73	18.78
18	5834	5784	50	99.14%	0.86%	105012	99948	5064	17.38%	60.70%	43.32	19.74
19	5834	5795	39	99.33%	0.67%	110846	105743	5103	18.39%	61.17%	42.78	20.72
20	5834	5790	44	99.25%	0.75%	110846	111533	5147	19.39%	61.69%	42.30	21.67

Testing Statistics for Neural Network

Testing	# Records	# Goods	# Bads	Fraud Rate
	250053	246394	3659	0.014632898

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	%Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	2500	652	1848	26.08%	73.92%	2500	652	1848	0.26%	50.51%	50.24	0.35
2	2500	2415	85	96.60%	3.40%	5000	3067	1933	1.24%	52.83%	51.58	1.59
3	2500	2478	22	99.12%	0.88%	7500	5545	1955	2.25%	53.43%	51.18	2.84
4	2500	2483	17	99.32%	0.68%	10000	8028	1972	3.26%	53.89%	50.64	4.07
5	2500	2484	16	99.36%	0.64%	12500	10512	1988	4.27%	54.33%	50.07	5.29
6	2500	2477	23	99.08%	0.92%	15000	12989	2011	5.27%	54.96%	49.69	6.46
7	2500	2480	20	99.20%	0.80%	17500	15469	2031	6.28%	55.51%	49.23	7.62
8	2500	2479	21	99.16%	0.84%	20000	17948	2052	7.28%	56.08%	48.80	8.75
9	2500	2478	22	99.12%	0.88%	22500	20426	2074	8.29%	56.68%	48.39	9.85
10	2500	2486	14	99.44%	0.56%	25000	22912	2088	9.30%	57.06%	47.77	10.97
11	2500	2485	15	99.40%	0.60%	27500	25397	2103	10.31%	57.47%	47.17	12.08
12	2500	2478	22	99.12%	0.88%	30000	27875	2125	11.31%	58.08%	46.76	13.12
13	2500	2480	20	99.20%	0.80%	32500	30355	2145	12.32%	58.62%	46.30	14.15
14	2500	2479	21	99.16%	0.84%	35000	32834	2166	13.33%	59.20%	45.87	15.16
15	2500	2477	23	99.08%	0.92%	37500	35311	2189	14.33%	59.83%	45.49	16.13
16	2500	2483	17	99.32%	0.68%	40000	37794	2206	15.34%	60.29%	44.95	17.13
17	2500	2473	27	98.92%	1.08%	42500	40267	2233	16.34%	61.03%	44.69	18.03
18	2500	2479	21	99.16%	0.84%	45000	42746	2254	17.35%	61.60%	44.25	18.96
19	2500	2479	21	99.16%	0.84%	47500	45225	2275	18.35%	62.18%	43.82	19.88
20	2500	2487	13	99.48%	0.52%	47500	47712	2288	19.36%	62.53%	43.17	20.85

OOT Statistics for Neural Network

OOT	# Records	# Goods	# Bads	Fraud Rate
	166493	164107	2386	0.014330933

	Bin Statistics					Cumulative Statistics						
Population Bin %	# Records	# Goods	# Bads	%Goods	% Bads	Total # Records	Cumulative Goods	Cumulative Bads	% Goods	% Bads (FDR)	KS	FPR
1	1665	505	1160	30.33%	69.67%	1665	505	1160	0.31%	48.62%	48.31	0.44
2	1665	1602	63	96.22%	3.78%	3330	2107	1223	1.28%	51.26%	49.97	1.72
3	1665	1635	30	98.20%	1.80%	4995	3742	1253	2.28%	52.51%	50.23	2.99
4	1665	1652	13	99.22%	0.78%	6660	5394	1266	3.29%	53.06%	49.77	4.26
5	1665	1656	9	99.46%	0.54%	8325	7050	1275	4.30%	53.44%	49.14	5.53
6	1665	1657	8	99.52%	0.48%	9990	8707	1283	5.31%	53.77%	48.47	6.79
7	1665	1652	13	99.22%	0.78%	11655	10359	1296	6.31%	54.32%	48.00	7.99
8	1665	1650	15	99.10%	0.90%	13320	12009	1311	7.32%	54.95%	47.63	9.16
9	1665	1650	15	99.10%	0.90%	14985	13659	1326	8.32%	55.57%	47.25	10.30
10	1665	1654	11	99.34%	0.66%	16650	15313	1337	9.33%	56.04%	46.70	11.45
11	1665	1653	12	99.28%	0.72%	18315	16966	1349	10.34%	56.54%	46.20	12.58
12	1665	1660	5	99.70%	0.30%	19980	18626	1354	11.35%	56.75%	45.40	13.76
13	1665	1654	11	99.34%	0.66%	21645	20280	1365	12.36%	57.21%	44.85	14.86
14	1665	1649	16	99.04%	0.96%	23310	21929	1381	13.36%	57.88%	44.52	15.88
15	1665	1648	17	98.98%	1.02%	24975	23577	1398	14.37%	58.59%	44.22	16.86
16	1665	1653	12	99.28%	0.72%	26640	25230	1410	15.37%	59.09%	43.72	17.89
17	1665	1648	17	98.98%	1.02%	28305	26878	1427	16.38%	59.81%	43.43	18.84
18	1665	1650	15	99.10%	0.90%	29970	28528	1442	17.38%	60.44%	43.05	19.78
19	1665	1650	15	99.10%	0.90%	31635	30178	1457	18.39%	61.06%	42.68	20.71
20	1665	1649	16	99.04%	0.96%	31635	31827	1473	19.39%	61.74%	42.34	21.61

Conclusion

Using the applications dataset containing 1,000,000 records and 10 original fields (including fraud labels), our group was able to develop a fraud detection algorithm using supervised learning methods.

The process began with understanding the data at hand, by plotting the distributions for the original fields, and cleaning the data by treating frivolous field values and filling in leading zeros to allow for consistency across the fields. Since the data is synthetic, based on real application data characteristics, there were no missing values and therefore imputation was not required. Next, 253 candidate variables were created using calculations for velocity, relative velocity, days since last seen and risk for day of week based on the original field values and combinations of these fields, over a series of different time periods.

After these variables were created, the data was standardized through z-scaling, with the last two months of the year (November and December) then removed for out of time validation (OOT). The number of variables was then cut in half, by using both univariate Kolmogorov-Smirnov (KS) and univariate Fraud Detection Rate (FDR) filters. The candidate variables were then further reduced to the top 27, using recursive feature elimination with cross-validation, or RFECV.

The data was then randomly split to contain 70% training data and 30% testing data and fed to classification models. Logistic regression was used as a simple baseline linear model, before implementing Boosted Tree, Random Forest, and Neural Network non-linear models. In the end, the model that performed the best after fine-tuning of parameters was Neural Network with an average FDR of **0.526** at 3% for **testing** and average FDR of **0.525** at 3% for **OOT**. This suggests that ~52% of the frauds are being caught in 3% of the population.

If given additional time, the following steps could have been taken to develop a stronger algorithm:

1. Creating hundreds of more candidate variables, which may result in stronger, more valuable inputs to be used in the model algorithms.
2. Exploring further non-linear models like SVM to identify a different model that may perform at a higher level of accuracy.
3. Performing more combinations of parameters for fine-tuning to get the most optimal results.

Appendix 1.1: Data Quality Report

The following Data Quality Report (DQR) summarizes the product application data and provides basic information about each of the features in the given dataset.

1. Data Description - The dataset provided is a synthesized labeled data for fraud that captures the essence of the original data.

Dataset Name	Product Application Data
Data Source	Synthetic Data
Time Period	2016 Full Year
# of Columns	10
# of Records	1,000,000

2. Summary - The overall dataset has 10 categorical fields (Including a date field) present. The detailed description is provided below:

2.1 Categorical Summary

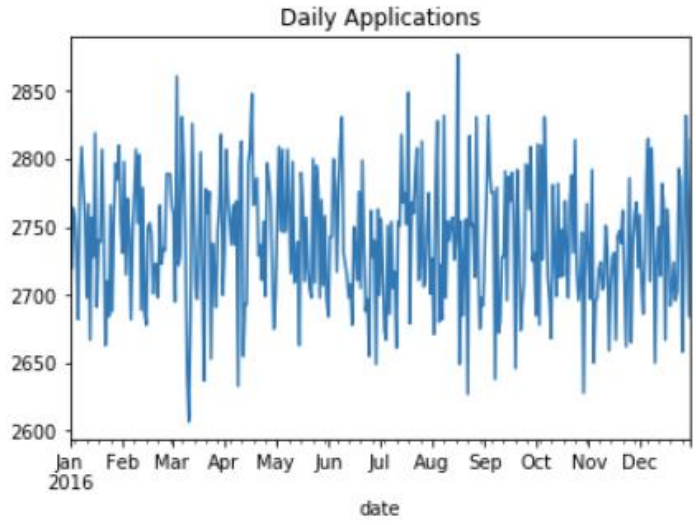
Field Name	# of Records With a Value	% Populated	# Unique Values	Most Common Field Value
record	1000000	100%	1000000	NA
Date	1000000	100%	365	20160816
ssn	1000000	100%	835819	999999999
firstname	1000000	100%	78136	EAMSTRMT
lastname	1000000	100%	177001	ERJSAXA
address	1000000	100%	828774	123 MAIN ST
zip5	1000000	100%	26370	68138
dob	1000000	100%	42673	19070626
homephone	1000000	100%	28244	999999999
fraud_label	1000000	100%	2	0

3. Data Field Exploration

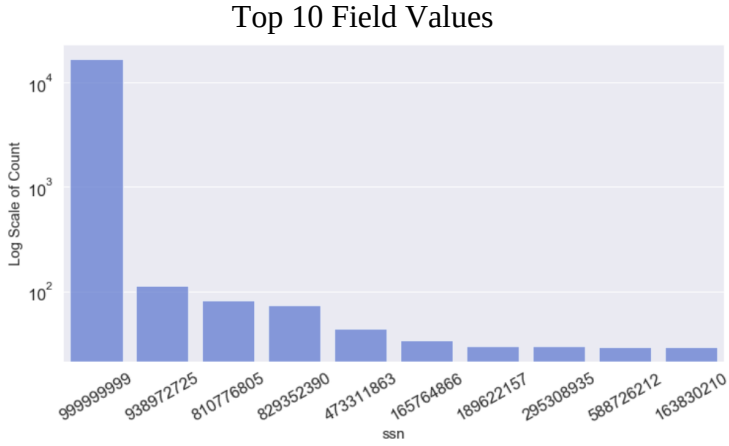
Below is a description of all the fields.

Field1
Field Name: record
Description: Unique identifier of each data record

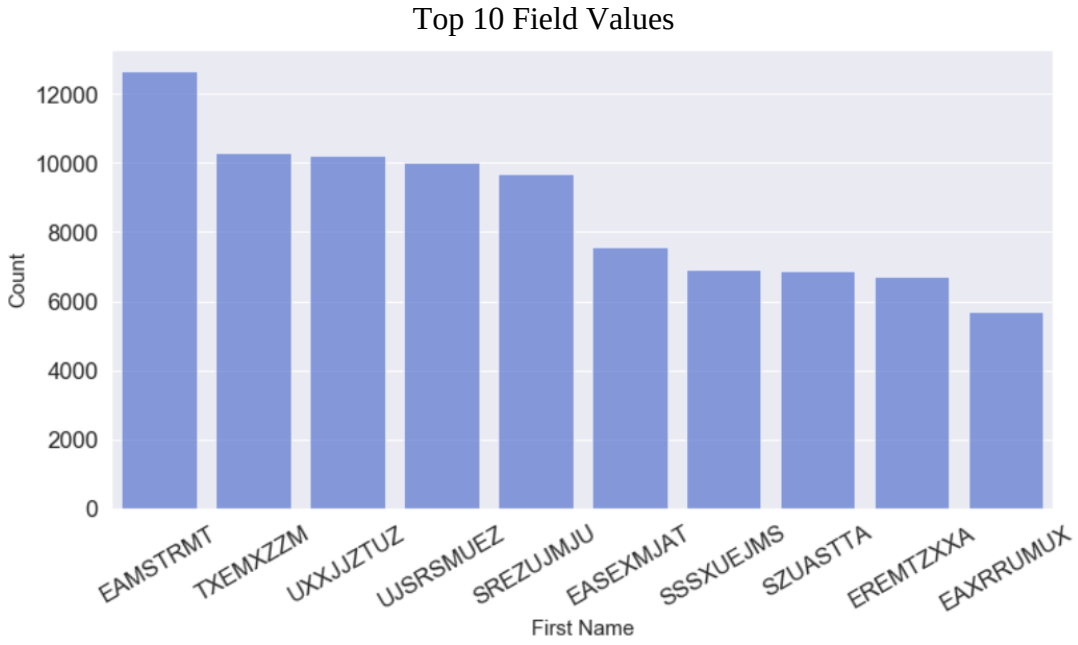
Field2
Field Name: date (Type: Categorical (Date/Time))
Description: Consists of date values for the entire year of 2016 (29th Feb data replaced with 28th Feb values since the data is missing)



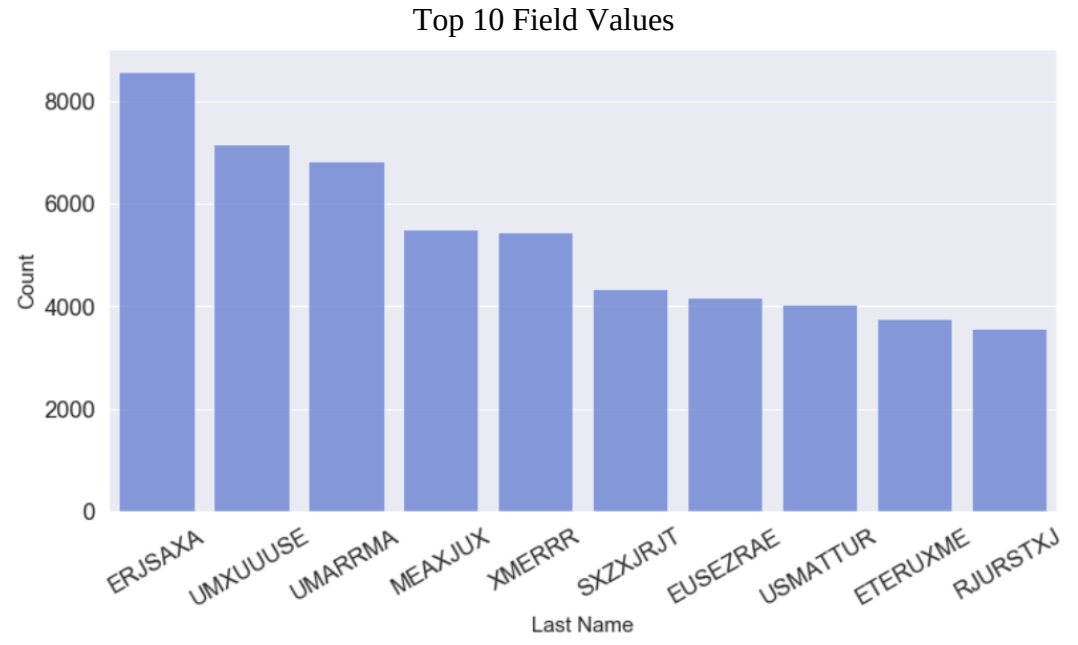
Field3
Field Name: ssn (Type: Categorical)
Description: Social Security Number of the applicant



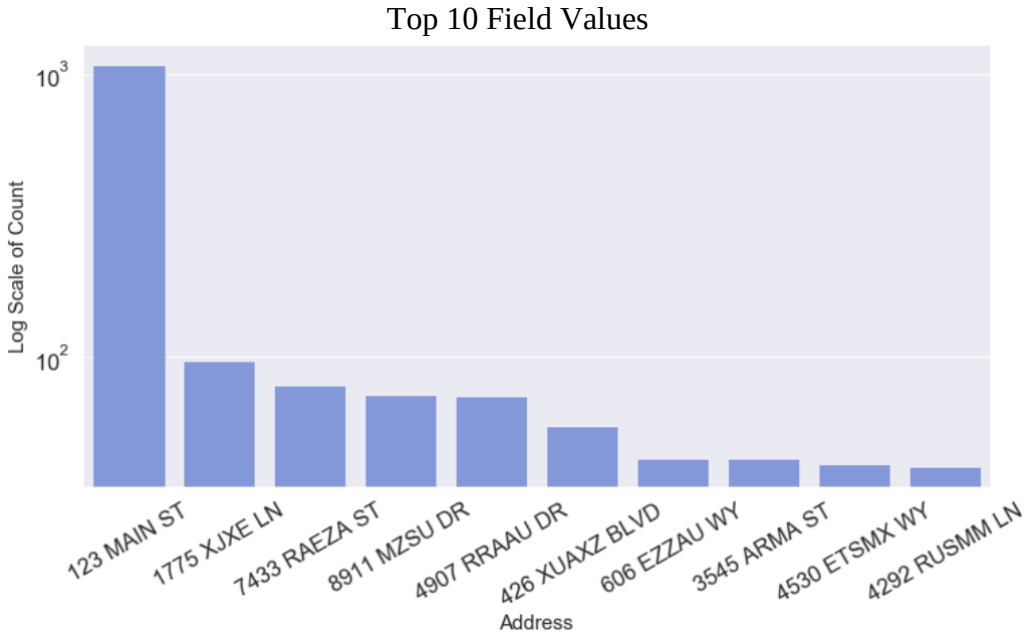
Field4
Field Name: firstname (Type: Categorical)
Description: First Name of the applicant



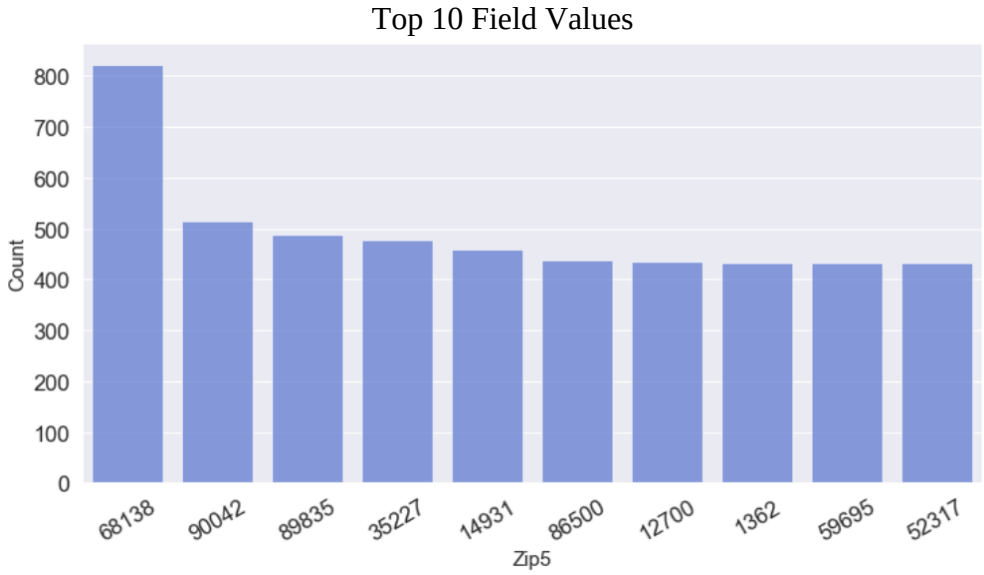
Field5
Field Name: lastname (Type: Categorical)
Description: Last Name of the applicant



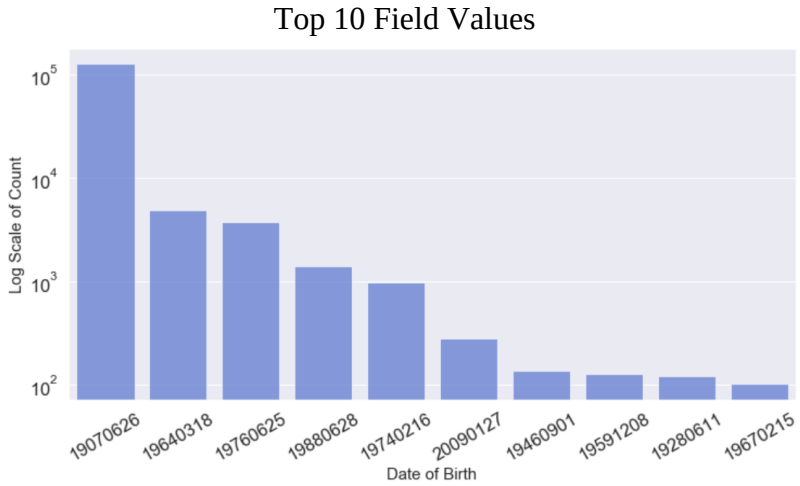
Field6
Field Name: address (Type: Categorical)
Description: Address of the applicant



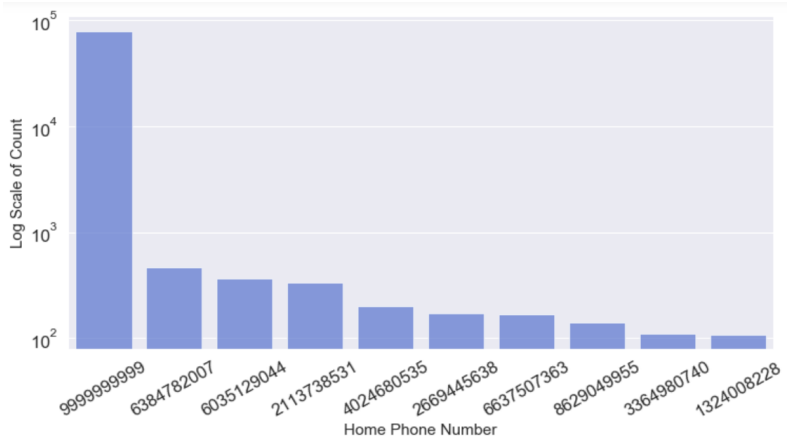
Field7
Field Name: zip5 (Type: Categorical)
Description: Zip code of the applicant. Any zip code with less than 5 digits is assumed to be having leading zeros.



Field8
Field Name: dob (Type: Categorical)
Description: Date of Birth of the applicant



Field9
Field Name: homephone (Type: Categorical)
Description: Home Phone number of the applicant



Field10
Field Name: fraud_label (Type: Categorical)
Description: Describes if the person is considered as fraud or not

Fraud Label	Count
0	985607
1	14393

Appendix 1.2: List of Candidate Variables

#	Candidate Variables	#	Candidate Variables
1	‘vel_ssn_1’	25	‘vel_namedob_14’
2	‘vel_ssn_3’	26	‘vel_namedob_30’
3	‘vel_ssn_7’	27	‘vel_namedob_90’
4	‘vel_ssn_14’	28	‘vel_namedob_180’
5	‘vel_ssn_30’	29	‘vel_fulladdress_1’
6	‘vel_ssn_90’	30	‘vel_fulladdress_3’
7	‘vel_ssn_180’	31	‘vel_fulladdress_7’
8	‘vel_homephone_1’	32	‘vel_fulladdress_14’
9	‘vel_homephone_3’	33	‘vel_fulladdress_30’
10	‘vel_homephone_7’	34	‘vel_fulladdress_90’
11	‘vel_homephone_14’	35	‘vel_fulladdress_180’
12	‘vel_homephone_30’	36	‘vel_ssnfulladdress_1’
13	‘vel_homephone_90’	37	‘vel_ssnfulladdress_3’
14	‘vel_homephone_180’	38	‘vel_ssnfulladdress_7’
15	‘vel_name_1’	39	‘vel_ssnfulladdress_14’
16	‘vel_name_3’	40	‘vel_ssnfulladdress_30’
17	‘vel_name_7’	41	‘vel_ssnfulladdress_90’
18	‘vel_name_14’	42	‘vel_ssnfulladdress_180’
19	‘vel_name_30’	43	‘vel_ssnnamedob_1’
20	‘vel_name_90’	44	‘vel_ssnnamedob_3’
21	‘vel_name_180’	45	‘vel_ssnnamedob_7’
22	‘vel_namedob_1’	46	‘vel_ssnnamedob_14’
23	‘vel_namedob_3’	47	‘vel_ssnnamedob_30’
24	‘vel_namedob_7’	48	‘vel_ssnnamedob_90’

49	‘vel_ssnnamedob_180’	75	‘vel_namedobphone_30’
50	‘vel_ssnphone_1’	76	‘vel_namedobphone_90’
51	‘vel_ssnphone_3’	77	‘vel_namedobphone_180’
52	‘vel_ssnphone_7’	78	‘vel_firstnamessn_1’
53	‘vel_ssnphone_14’	79	‘vel_firstnamessn_3’
54	‘vel_ssnphone_30’	80	‘vel_firstnamessn_7’
55	‘vel_ssnphone_90’	81	‘vel_firstnamessn_14’
56	‘vel_ssnphone_180’	82	‘vel_firstnamessn_30’
57	‘vel_namedobfulladdress_1’	83	‘vel_firstnamessn_90’
58	‘vel_namedobfulladdress_3’	84	‘vel_firstnamessn_180’
59	‘vel_namedobfulladdress_7’	85	‘vel_lastnamessn_1’
60	‘vel_namedobfulladdress_14’	86	‘vel_lastnamessn_3’
61	‘vel_namedobfulladdress_30’	87	‘vel_lastnamessn_7’
62	‘vel_namedobfulladdress_90’	88	‘vel_lastnamessn_14’
63	‘vel_namedobfulladdress_180’	89	‘vel_lastnamessn_30’
64	‘vel_fulladdressphone_1’	90	‘vel_lastnamessn_90’
65	‘vel_fulladdressphone_3’	91	‘vel_lastnamessn_180’
66	‘vel_fulladdressphone_7’	92	‘vel_namephone_1’
67	‘vel_fulladdressphone_14’	93	‘vel_namephone_3’
68	‘vel_fulladdressphone_30’	94	‘vel_namephone_7’
69	‘vel_fulladdressphone_90’	95	‘vel_namephone_14’
70	‘vel_fulladdressphone_180’	96	‘vel_namephone_30’
71	‘vel_namedobphone_1’	97	‘vel_namephone_90’
72	‘vel_namedobphone_3’	98	‘vel_namephone_180’
73	‘vel_namedobphone_7’	99	‘vel_namefulladdress_1’
74	‘vel_namedobphone_14’	100	‘vel_namefulladdress_3’

#	Candidate Variables	#	Candidate Variables
101	‘vel_namefulladdress_7’	127	‘relv_ssn_3’
102	‘vel_namefulladdress_14’	128	‘relv_ssn_7’
103	‘vel_namefulladdress_30’	129	‘relv_ssn_14’
104	‘vel_namefulladdress_90’	130	‘relv_ssn_30’
105	‘vel_namefulladdress_180’	131	‘relv_ssn_90’
106	‘vel_namezip_1’	132	‘relv_ssn_180’
107	‘vel_namezip_3’	133	‘relv_homephone_3’
108	‘vel_namezip_7’	134	‘relv_homephone_7’
109	‘vel_namezip_14’	135	‘relv_homephone_14’
110	‘vel_namezip_30’	136	‘relv_homephone_30’
111	‘vel_namezip_90’	137	‘relv_homephone_90’
112	‘vel_namezip_180’	138	‘relv_homephone_180’
113	‘vel_namephonefulladdress_1’	139	‘relv_name_3’
114	‘vel_namephonefulladdress_3’	140	‘relv_name_7’
115	‘vel_namephonefulladdress_7’	141	‘relv_name_14’
116	‘vel_namephonefulladdress_14’	142	‘relv_name_30’
117	‘vel_namephonefulladdress_30’	143	‘relv_name_90’
118	‘vel_namephonefulladdress_90’	144	‘relv_name_180’
119	‘vel_namephonefulladdress_180’	145	‘relv_namedob_3’
120	‘vel_namessn_1’	146	‘relv_namedob_7’
121	‘vel_namessn_3’	147	‘relv_namedob_14’
122	‘vel_namessn_7’	148	‘relv_namedob_30’
123	‘vel_namessn_14’	149	‘relv_namedob_90’
124	‘vel_namessn_30’	150	‘relv_namedob_180’
125	‘vel_namessn_90’	151	‘relv_fulladdress_3’
126	‘vel_namessn_180’	152	‘relv_fulladdress_7’

153	‘relv_fulladdress_14’	180	‘relv_namedobfulladdress_180’
154	‘relv_fulladdress_30’	181	‘relv_fulladdressphone_3’
155	‘relv_fulladdress_90’	182	‘relv_fulladdressphone_7’
156	‘relv_fulladdress_180’	183	‘relv_fulladdressphone_14’
157	‘relv_ssnfulladdress_3’	184	‘relv_fulladdressphone_30’
158	‘relv_ssnfulladdress_7’	185	‘relv_fulladdressphone_90’
159	‘relv_ssnfulladdress_14’	186	‘relv_fulladdressphone_180’
160	‘relv_ssnfulladdress_30’	187	‘relv_namedobphone_3’
161	‘relv_ssnfulladdress_90’	188	‘relv_namedobphone_7’
162	‘relv_ssnfulladdress_180’	189	‘relv_namedobphone_14’
163	‘relv_ssnnamedob_3’	190	‘relv_namedobphone_30’
164	‘relv_ssnnamedob_7’	191	‘relv_namedobphone_90’
165	‘relv_ssnnamedob_14’	192	‘relv_namedobphone_180’
166	‘relv_ssnnamedob_30’	193	‘relv_firstnamessn_3’
167	‘relv_ssnnamedob_90’	194	‘relv_firstnamessn_7’
168	‘relv_ssnnamedob_180’	195	‘relv_firstnamessn_14’
169	‘relv_ssnphone_3’	196	‘relv_firstnamessn_30’
170	‘relv_ssnphone_7’	197	‘relv_firstnamessn_90’
171	‘relv_ssnphone_14’	198	‘relv_firstnamessn_180’
172	‘relv_ssnphone_30’	199	‘relv_lastnamessn_3’
173	‘relv_ssnphone_90’	200	‘relv_lastnamessn_7’
174	‘relv_ssnphone_180’	201	'relv_lastnamessn_14'
175	‘relv_namedobfulladdress_3’	202	'relv_lastnamessn_30'
176	‘relv_namedobfulladdress_7’	203	'relv_lastnamessn_90'
177	‘relv_namedobfulladdress_14’	204	'relv_lastnamessn_180'
178	‘relv_namedobfulladdress_30’	205	'relv_namephone_3'
179	‘relv_namedobfulladdress_90’	206	'relv_namephone_7'

#	Candidate Variables	#	Candidate Variables
207	'relv_namephone_14'	231	'relv_namessn_14'
208	'relv_namephone_30'	232	'relv_namessn_30'
209	'relv_namephone_90'	233	'relv_namessn_90'
210	'relv_namephone_180'	234	'relv_namessn_180'
211	'relv_namefulladdress_3'	235	'dayssince_ssn'
212	'relv_namefulladdress_7'	236	'dayssince_homephone'
213	'relv_namefulladdress_14'	237	'dayssince_name'
214	'relv_namefulladdress_30'	238	'dayssince_namedob'
215	'relv_namefulladdress_90'	239	'dayssince_fulladdress'
216	'relv_namefulladdress_180'	240	'dayssince_ssnfulladdress'
217	'relv_namezip_3'	241	'dayssince_ssnnamedob'
218	'relv_namezip_7'	242	'dayssince_ssnphone'
219	'relv_namezip_14'	243	'dayssince_namedobfulladdress'
220	'relv_namezip_30'	244	'dayssince_fulladdressphone'
221	'relv_namezip_90'	245	'dayssince_namedobphone'
222	'relv_namezip_180'	246	'dayssince_firstnamessn'
223	'relv_namephonefulladdress_3'	247	'dayssince_lastnamessn'
224	'relv_namephonefulladdress_7'	248	'dayssince_namephone'
225	'relv_namephonefulladdress_14'	249	'dayssince_namefulladdress'
226	'relv_namephonefulladdress_30'	250	'dayssince_namezip'
227	'relv_namephonefulladdress_90'	251	'dayssince_namephonefulladdress'
228	'relv_namephonefulladdress_180'	252	'dayssince_namessn'
229	'relv_namessn_3'	253	'risk_dayofweek'
230	'relv_namessn_7'		

Appendix 1.3: 27 Best Variables

#	Candidate Variables
1	'dayssince_fulladdress'
2	'vel_fulladdress_30'
3	'vel_fulladdress_14'
4	'vel_fulladdress_7'
5	'vel_fulladdress_3'
6	'relv_fulladdress_14'
7	'relv_fulladdress_30'
8	'relv_fulladdress_60'
9	'vel_fulladdress_1'
10	'dayssince_fulladdressphone'
11	'vel_fulladdressphone_30'
12	'dayssince_namedob'
13	'vel_namedob_30'
14	'dayssince_ssn'
15	'vel_ssn_30'
16	'dayssince_ssnnamedob'
17	'vel_ssnnamedob_30'
18	'vel_fulladdressphone_60'
19	'dayssince_namessn'
20	'vel_namessn_30'
21	'vel_namedob_60'
22	'vel_ssnnamedob_60'
23	'vel_fulladdressphone_14'
24	'vel_namessn_60'
25	'vel_namedob_14'
26	'vel_ssndob_14'
27	'vel_ssn_14'